2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation

Alejandro R. Mosteo and Luis Montano

Abstract— **Auction based techniques are a highly successful tool used for multi-robot task allocation. However, theoretical performance and a proper taxonomy of optimization objectives have remained scarce until recent studies. Implementations from different authors have not been compared in common grounds and in light of these recent findings. In this paper we address this lack of comparative experimentation, providing simulation results on a large real life based scenario and in random worlds. Two intuitive optimization objectives, minimum total resource usage and minimum total time, are evaluated in object searching missions. A method for flexible tailoring of the bidding rules is presented and new insight is gained on the effect of using hybrid criteria for the optimization objective.**

I. INTRODUCTION

In recent years, researchers have moved towards multi-robot systems in order to solve problems in less time, more efficiently and with higher reliability. In some cases the robots act independently of each other, but the use of explicit coordination offers potential advantages in efficiency and flexibility. However, this use of explicit cooperation introduces a new challenge: determining the optimal utilization of team members. The allocation of tasks to robots has been shown several times to be an $\mathcal{NP}$-complete problem [1]. For this reason, approximated methods have been used extensively.

We focus on auction based methods [2], which are one of the most successfully tested solutions. Auction methods are based on the exchange of bids amongst robots; bids are offered for the tasks and the winner of a task takes responsibility to carry it out. Auction methods are decentralized in nature, since no central authority controls the bidding process. They are also efficient bandwidth-wise because all relevant information is synthesized and exchanged by means of such bids, which usually take the form of single scalars. Experimental results have shown that they perform very well in real life, often far above their worst case bounds when known. For all these reasons they have been the subject of abundant research.

Numerous implementations have been described ([3]–[6] to cite a few) but efforts for strong mathematical characterization of its efficiency have been scarcer [1], [7], [8]. Also, although they are usually presented as flexible general task allocation methods not particularly bounded to a definite problem, the effects of different optimization criteria are seldom thoroughly analyzed. We are not aware of comparisons on performance for the different implementations on a common ground. In this paper we aim to fill this gap with simulation runs over a model of a real world large building in a mission for object searching. Also, randomized experiments are used to extract trends not confined to this scenario. For flexible mission objective tailoring, we use a technique that we call *criterion descriptors* [9]. We explore the effects of the different criteria on the team performance, and highlight the interesting good performance of hybrid objectives. To the best of our knowledge, this has not been previously observed in the context of auction allocation.

We will start examining the theoretical models underlying the auction process in section II. Next, we present in section III the auction methods we have implemented. We follow with simulations and its discussion in section IV, to finally conclude with a summary of the most relevant results.

II. OPTIMIZATION CRITERIA

Auction methods can be modelled for minimization (cost based [7]) or for maximization (utility based [10]). In this paper we will adopt the first approach; hence, the lowest bid is the best one and cost will be the used metric.

We focus on two intuitive and useful optimization criteria for a robotic team. The MINSUM objective aims to minimize the total usage of resources (sum of individual costs). It is the most commonly one found in auction based research, because it naturally arises in market-mimicking varieties where agents try to minimize their costs in order to maximize their own gains, and the team reward is defined as the sum of individual rewards [10].

Other interesting criterion, less commonly found, is the MINMAX one, where the objective is to minimize the cost of the worst performing robot. This has a direct translation to finding the shortest timespan for a given mission [5].

A detailed mathematical analysis of these criteria is found in [7] in the context of auctions for vehicle routing. Since many robotic endeavors consist of visiting places to there perform tasks of interest (analysis of ground, deployment of sensors, surveillance), this is of direct relevance.

More formally, let us define $\mathcal{R} = \{r_1, .., r_n\}$ as the set of robots in the team and $\mathcal{T} = \{t_1, .., t_m\}$ as the set of tasks to be executed. An allocation $\mathcal{A} = \{a_1, .., a_n\}$ is a partition of $\mathcal{T}$ where a_i is an ordered list of tasks assigned to be executed by robot r_i. That is, we are using a *sequential execution model* where robots are capable of executing a task at a time. Let $C(r_i, a_i)$ be the cost robot r_i incurs when executing the ordered sequence a_i. Hence, the costs for the

This work was funded by the Spanish MCYT-FEDER projects DPI2003-07986 and DPI2006-07928, and EU project IST-1-045062-URUS-STP.

Authors are with Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, and Instituto de Investigación en Ingeniería de Aragón, I3A. {amosteo,montano}@unizar.es

two presented team objectives are

$$C_{\mathrm{MM}} = \max_i C(r_i, a_i)$$
$$C_{\mathrm{MS}} = \sum_i C(r_i, a_i)$$

where MM and MS are abbreviations for MINMAX and MINSUM respectively.

With these definitions, each of the two presented criteria aim to find the optimal allocation $\mathcal{A}^O$ such as

$$\mathcal{A}^O_{\mathrm{MM}} = \arg\min_{\mathcal{A}} C_{\mathrm{MM}}$$
$$\mathcal{A}^O_{\mathrm{MS}} = \arg\min_{\mathcal{A}} C_{\mathrm{MS}}$$

In [7], bidding rules are defined for these two criteria, in a framework for optimality analysis. Performance bounds are given in the form of competitivity ratios ($\frac{\text{Worst case cost}}{\text{Optimal cost}}$) for the studied bidding rules. We have identified that several auction implementations use at least the MINSUM bid construction rule, even if exact timing details differ; and we ourselves have used the MINMAX bids in past work. For the flexible switching between both criteria we have defined in [9] *criterion descriptors* as a tuple $(\omega_{\mathrm{MM}}, \omega_{\mathrm{MS}})$, with ω_{MM} being a weight factor for the MINMAX cost, and ω_{MS} a weight factor for the MINSUM cost. The team allocation cost is computed as

$$C(\mathcal{A}) = \omega_{\mathrm{MM}} C_{\mathrm{MM}} + \omega_{\mathrm{MS}} C_{\mathrm{MS}}$$

A robotic team can use a unique bidding rule parametrized with a criterion descriptor for flexible switching between objectives. Let us assume t is a new task being auctioned. Let us denote by a_i the assignation for r_i before the task is added to its plan, and by a'_i the best assignation r_i is able to find including t in its plan. In [7] is shown that the bid b_i of r_i for each objective is to be:

$$\begin{array}{llll} \text{MINMAX}: & b_i = & C_{Ti} = & C(r_i, a'_i) \\ \text{MINSUM}: & b_i = & C_{\Delta i} = & C(r_i, a'_i) - C(r_i, a_i) \end{array}$$

where C_{Ti} stands for *total cost* and $C_{\Delta i}$ stands for *incremental cost*. Both quantities can be maintained and computed by a robot in constant time, so without performance losses we propose the following unique rule:

$$b_i = \omega_{\mathrm{MM}} C_{Ti} + \omega_{\mathrm{MS}} C_{\Delta i}$$

where a criterion descriptor is used to tailor the desired optimization criterion. In this paper we will experiment with the following descriptors:

MINMAX:	$(1.0, 0.0)$	MINSUM:	$(0.0, 1.0)$
MINTIM:	$(1.0, \delta)$	MINMIX:	$(1.0, 1.0)$

We name the descriptors after the objective they pursue. Here appear two notable additions, that we call hybrid for its use of both C_{MM} and C_{MS}: MINTIM (where δ represents a small real[1]) is our attempt to provide minimum time but also improving the costs of robots not involved in the critical worst cost of a pure MINMAX optimization [9]. The idea is that the dominating cost is still the C_{MM} one, but the C_{MS} scaled down cost is used to prefer less costly assignations for the remaining robots. On the other hand, MINMIX uses a balanced combination of both costs in an attempt to achieve good performance for both objectives. The exact meaning in physical units is not clear, but our interest is to properly ascertain if the saying *"Jack of all trades, master of none"* applies here or, in the contrary, it is a useful alternative.

Note that all descriptors will be evaluated in terms of pure C_{MM} and C_{MS} performance, because these are the magnitudes with real meaning. In our discussions we will often refer to cost as *time* or *resources*. When the rationale is MINMAX, time is a common measure of cost and refers to the mission timespan. When the rationale is MINSUM, resources could represent the sum of distance traveled or power (batteries, fuel) consumed.

[1] *E.g.* 0,00001.

III. IMPLEMENTED ALGORITHMS

In this section we present the relevant auctioning techniques we have implemented for testing. We are particularly interested in techniques with proved performance bounds or, lacking this, high popularity. For this initial testbed we have preferred methods that are of similar complexity, specifically in the order of the one with known performance bounds. We have thus not tested complex methods like combinatorial or hierarchical auctions [3]. The bidding rule is the one explained except when noted, and usually what changes is the timing details.

A. Parallel auctions with performance bounds

This implementation follows the full directions given in [7] for auctions with proved performance bounds when the triangle inequality holds, *e.g.* in routing problems. We call it LAGO in reference to the first author. In these auctions, initially no tasks are allocated. In each round a task is allocated this way: Each agent in parallel computes bids for all the remaining tasks, using an insertion heuristic. The lowest bid of all robots wins and the appropriate task is awarded. After m rounds, all tasks in $\mathcal{T}$ have been awarded and the auction ends. In our implementation robots and tasks have an implicit order, used to resolve ties.

We also test a variant identified as LAGOTAIL. Now, a task can only be added to the tail of a robot plan (instead of considering all possible insertions). This kind of auction does not have any quality bounds that we know, but is tested here because it is extremely cheap to be computed when robots have very long task lists.

B. Single item auctions

In this case, agents do not consider bids for all tasks in each round, but only for a single task that in our implementation is chosen at random. Again, m auction runs are performed until no task remains unassigned. We call this SINGLE in our tables. SINGLEEXT, in turn, refers to the allocation found after prolonging for one extra minute the auction process. During this extra time, in each round

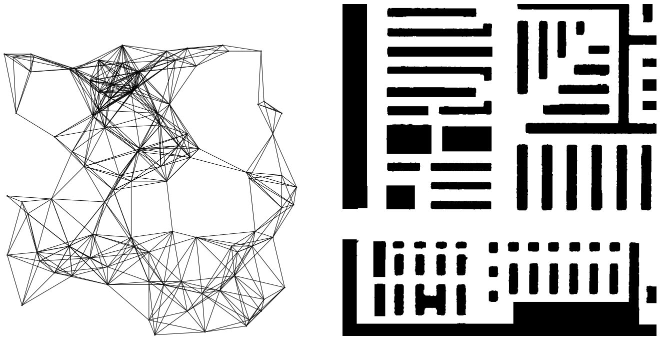

Fig. 1. Left: an example of randomly generated graph for the randomized worlds. Right: a quarter of the floorplan used for the large grid simulation.

some robot auctions one of its already owned tasks (see the accompanying video). In our implementation, auctions occur at a rate of approximately 60 auctions per second, so roughly 3600 additional rounds are performed. In real robotic teams, usually auctions are always running during mission execution.

This resembles mechanisms used when not all the tasks can be known in advance, or where tasks are generated by the robots during the mission. In this case, to prevent entering in conflicting bids, robots bid for a single task at a time. A possible example of this kind of implementation is [10].

C. *Allocation on availability*

This method is partially inspired in [6], which is considered the first auction based embedded implementation tested in real robots. Its singular characteristic is that robots can have just a task in their plan; when a new task is introduced, the best suited idle robot wins it via regular bidding and abstains of further bidding until it is idle again. At that point, it will choose among the unallocated tasks the one for which it has the lowest bid. (In [6], task reintroduction was not implemented.) In other words, this is an initial greedy allocation where the first robot to become idle executes the cheapest available task. We will refer to our implementation as FIFO (because the first robot to become idle is the first one to choose a task).

IV. Simulations and Comparatives

The first kind of scenario are graphs composed by a hundred randomly placed locations in a bounded area, with the restriction that not all locations are reachable from each other. See fig. 1 for an example. Robots start at a random node and all remaining nodes are goals that must be visited. This setup aims to be a generic testcase.

The second scenario is a real building floorplan (see fig. 1), modelled as a discrete grid of over 1200 cells (see fig. 2). Each cell adjacent to a wall is considered a goal task that must be visited by some robot[2]. Real examples of application comprehend: vehicle location in a parking, explosive sniffing in buildings, inventory maintenance with RFID[3] tags. Unity

[2]There are 1023 of these cells.

[3]Radio Frequency Identification

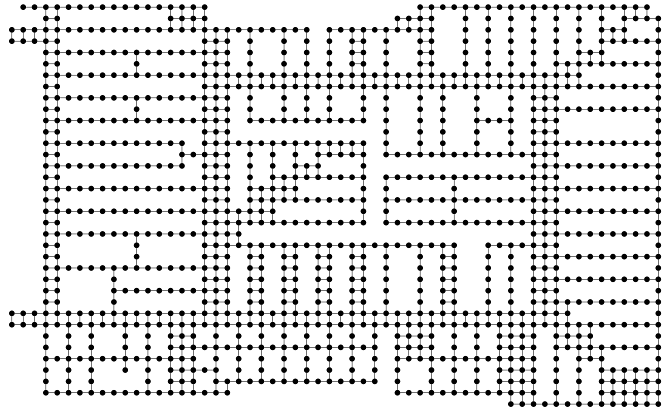

Fig. 2. Grid extracted from building floorplans. Each vertex is a reachable place. Vertices not completely surrounded by reachable space are goal places to be visited. The robots are initially positioned at the four leftmost vertices.

is the cost of traveling from cell to cell, thus implying holonomic robots. Cells have 4-vicinity as depicted in fig. 2. Robots travel using the shortest path in the grid graph and start in the four leftmost vertices (building entrances).

All simulations use four robots because this is the size of our real robotic team, with which we intend to carry real experimentation in the future in the real building.

Experiments in subsection IV-A aim to extract general trends using multiple random graphs, and to observe if these trends appear also in the grid map. Subsections IV-B and IV-C use the best solutions found in the grid map to evaluate mission variants.

A. *Costs of full allocations*

The objective in this experiment is to visit all goals in a map. For the random graphs we performed a hundred runs, each in a randomly generated world like the one shown in fig. 1. Average costs are shown in table I and fig. 3.

In the grid map (table II), only Single and SingleExt results are averages of a hundred runs. This is because their algorithms have a random step, while the others are deterministic for a given initial world state.

We have computed the allocations for all combinations of implementations and descriptors. The FIFO case is special, because its single task rule precludes the meaningful use of descriptors, so only one solution exists. In the grid map we also have computed the optimal MTSP[4] solution for the C_{MS} cost using the Concorde solver [11]. The leftmost column names the combinations, while the topmost row names the allocation evaluation (*i.e.* team performance). We give now some highlights on these results.

1) Descriptors: Clearly seen in fig. 3 is that any descriptor with nonzero ω_{MM} performs comparably well for the C_{MM} cost, with small differences in the 5% range. For the C_{MS} cost differences are smaller; MinSum is the best descriptor, MinMax and MinTim are comparably worse and MinMix is in middle ground.

[4]Multiple Traveling Salesmen Problem

TABLE I

MEAN COSTS OF FULL ALLOCATIONS IN RANDOM SCENARIOS

	Allocation evaluation			
	MinMax		MinSum	
Algorithm + Descriptor	C_{MM}	Ratio to best	C_{MS}	Ratio to best
Best found	228	1.00	816	1.00
Lago MinMax	262	1.15	943	1.16
Lago MinTim	**259**	**1.14**	935	1.15
Lago MinMix	272	1.19	903	1.11
Lago MinSum	463	2.03	**849**	**1.04**
LagoTail MinMax	**269**	**1.18**	962	1.18
LagoTail MinTim	270	1.18	971	1.19
LagoTail MinMix	272	1.19	904	1.11
LagoTail MinSum	387	1.69	**868**	**1.06**
Single MinMax	259	1.13	998	1.22
Single MinTim	260	1.14	1000	1.23
Single MinMix	**249**	**1.09**	932	1.14
Single MinSum	399	1.75	**847**	**1.04**
SingleExt MinMax	240	1.05	949	1.16
SingleExt MinTim	235	1.03	924	1.13
SingleExt MinMix	**228**	**1.00**	869	1.06
SingleExt MinSum	385	1.68	**816**	**1.00**
Fifo	320	1.40	1159	1.42

2) *Worst cases:* Observe that using the MinSum criterion causes a huge overcost for C_{MM} (in the range of 42% to 200%, within a same algorithm), whereas using a descriptor with nonzero ω_{MM} gives a worst overcost for C_{MS} in the 5%-18% range only.

The huge overcost of the MinSum-C_{MM} combination emanates from the fact that, when tasks are easily chainable to one another, is more resource-effective to just move a robot along a path than to move several of them. Thus, is possible for a single robot to have a high cost which is also a great percentage of the total team cost. See fig. 4 as example, where a robot is visiting only two goals. This would be aggravated in teams with more robots.

3) MinMix: This descriptor ranges from being the best one in many cases to a 5% overcost for C_{MM}, and has a penalty of 1%-10% for C_{MS}. From the previous points, it seems clear that using a criterion with ω_{MM} is better for general purpose robotic teams, since they perform very well for both C_{MM} and C_{MS} solutions. Furthermore, in our tests, MinMix wins over the other two ω_{MM} criteria in more cases. MinSum is only justified when the only relevant metric is C_{MS} (which leaves out missions involving time).

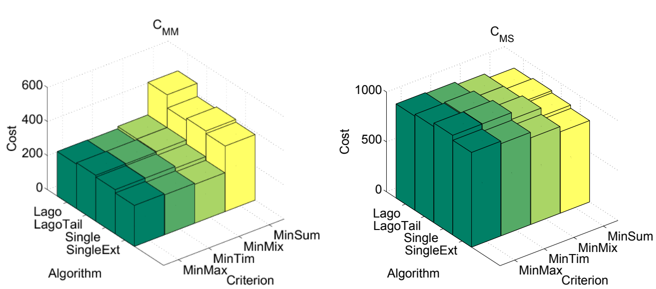

Fig. 3. Average C_{MM} and C_{MS} costs in the random worlds.

TABLE II

COSTS FOR THE LARGE GRID WORLD

	Allocation evaluation			
	MinMax		MinSum	
Algorithm + Descriptor	C_{MM}	Ratio to best	C_{MS}	Ratio to opt.
Optimal	**-**	**-**	1285	1.00
Best found	371	1.00	1455	1.13
Lago MinMax	449	1.21	1723	1.34
Lago MinTim	451	1.22	1789	1.39
Lago MinMix	**441**	**1.19**	1735	1.35
Lago MinSum	1331	3.59	**1709**	**1.33**
LagoTail MinMax	377	1.02	1497	1.16
LagoTail MinTim	411	1.11	1609	1.25
LagoTail MinMix	**371**	**1.00**	1467	1.14
LagoTail MinSum	560	1.51	**1455**	**1.13**
Single MinMax	458	1.23	1825	1.42
Single MinTim	452	1.22	1802	1.40
Single MinMix	**433**	**1.17**	1721	1.34
Single MinSum	1113	3.00	**1627**	**1.27**
SingleExt MinMax	448	1.21	1788	1.39
SingleExt MinTim	438	1.18	1748	1.36
SingleExt MinMix	**420**	**1.13**	1673	1.30
SingleExt MinSum	1115	3.00	**1610**	**1.25**
Fifo	491	1.32	1783	1.39
Optimal MTSP MinSum	675	1.82	1285	1.00

4) *Algorithms:* We observe that the SingleExt auctions are the best ones in most cases. This is due to the extra auction runs, as evidenced by the improvement over the Single results. This strongly suggests to use cost-bounded Lago auctions for the initial allocation when possible and continue with SingleExt rounds for the entire mission duration. Fifo is worst in all cases but when the least suited descriptor is used for an objective (*e.g.* MinSum for the C_{MM} cost).

5) *Anomalies:* Lago performs notably worse than LagoTail in the grid world, even if the latter algorithm is simpler. The runs in randomized worlds show that this is not a general trend.

6) *Bounds:* The algorithm with theoretical proved bounds, Lago, is well within these bounds in the grid run. Table III shows the solution costs compared against their theoretical bounds, as per [7]. Note that we know the optimal C_{MS} cost, but not the C_{MM} one, so its column reflects the ratio against the best known solution (showing thus a too optimistic value). Also note that all runs by the other algorithms are within these bounds too.

TABLE III

Lago $\frac{\text{sol. cost}}{\text{optimal}}$ ratios compared to their theoretical bounds for n robots.

	Team objective	
Bidding criterion	C_{MM}	C_{MS}
MinMax	$1.21 \leq 2n$	$1.34 \leq 2n$
MinSum	$3.59 \leq 2n$	$1.33 \leq 2$

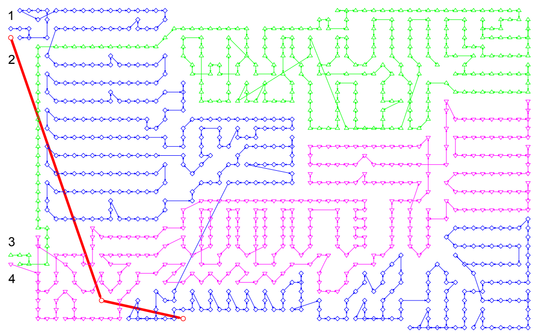

Fig. 4. The best MINSUM solution found via auctions. Note that the second robot from top (bolder) only performs two tasks.

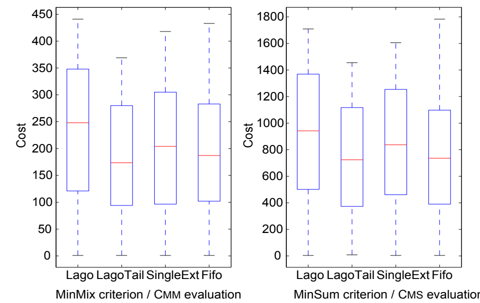

Fig. 6. The four auction algorithms compared. Left: Mission time evaluation. Right: Resource usage evaluation.

B. Single object finding

In this section we highlight some results on the mission of finding an object randomly placed in the grid world. Now we measure the cost until the object is found and not the total plan cost. The two box diagrams to the left in fig. 5 show the distribution of costs for the best C_{MM} and C_{MS} solutions found via auctions (LAGOTAIL-MINMIX and LAGOTAIL-MINSUM combinations respectively), over 1000 runs. When the team evaluation is to minimize time, the C_{MM} solution performs patently better. However, when the team evaluation is to minimize resource usage, we observe a similar performance from both solutions. This confirms the findings in the previous subsection about the convenience of descriptors with nonzero ω_{MM} weight.

The histograms to the right of that same figure reveal that the distribution that appears when repeating the mission 30.000 times is roughly uniform for the implemented algorithms, except for the FIFO case. We understand this inspecting fig. 6, which shows the distributions of each algorithm over 1000 runs. We see that all of them but FIFO have a uniform appearance, while FIFO has a lower median compared to its mean. This is caused by its greedy nature: FIFO tends to perform better toward the beginning, but this short-sightedness causes a higher cost at the tail of the plan, where far apart goals will remain. We will observe this again in the remaining experiments.

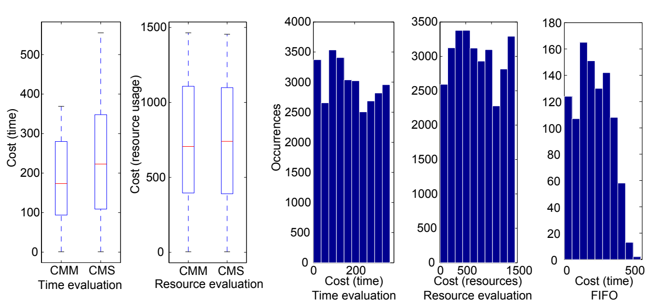

Fig. 5. Left: The best C_{MM} and C_{MS} solutions, evaluated for both mission time and total resources objectives. Median, quartiles and extreme values are shown. Right: histogram of cost for the best C_{MM} solution with time evaluation, C_{MS} solution with resource evaluation and FIFO with time evaluation.

C. Objects found in limited time

In this experiment we place a hundred objects in random places of the grid world, limit the mission time, and count how many of the objects are found. Three variations are presented:

1) C_{MM} vs. C_{MS}: We compare the best solution of each class, when the mission deadline is fixed to half the C_{MM} time (note that in half the C_{MS} time, the C_{MM} solution would almost finish the exploration). As shown by fig. 7, the shorter exploration timespan of C_{MM} allows to find more objects. Even if this seems evident, we consider this of relevance to multi-robot exploration where reward maximization is used: this gives a good ratio $\frac{\text{Area explored}}{\text{Distance traveled}}$, but exploration time can be highly suboptimal if our observations are confirmed in these problems.

2) Halved algorithm solution time: We test each algorithm taking half the time of their best solution as mission deadline. Thus, the algorithm with lowest median, FIFO, finds more objects (fig. 8a). This confirms the rationale argued in [6]: the instantly greedy FIFO strategy is best when there is no information on task arrival, since greedy allocation will give the lowest average cost per task. For infinite horizon problems with unbounded task arrival, FIFO would likely be the best algorithm.

3) Fixed global time: But our mission has a finite environment with finite objects. Now, all algorithms share the same deadline: half the time of the globally best known solution. This is a fairer comparison of absolute performance. As shown by fig. 8b, FIFO is no longer the best, since its solution is more costly. The short-term advantage is neglected by the

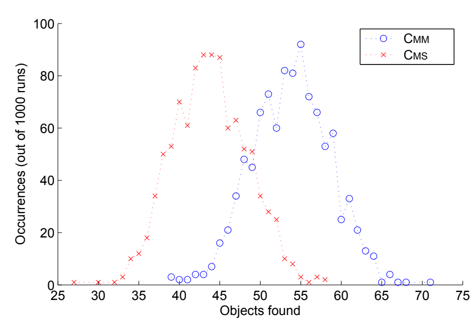

Fig. 7. Objects found by the best C_{MM} and C_{MS} allocations.

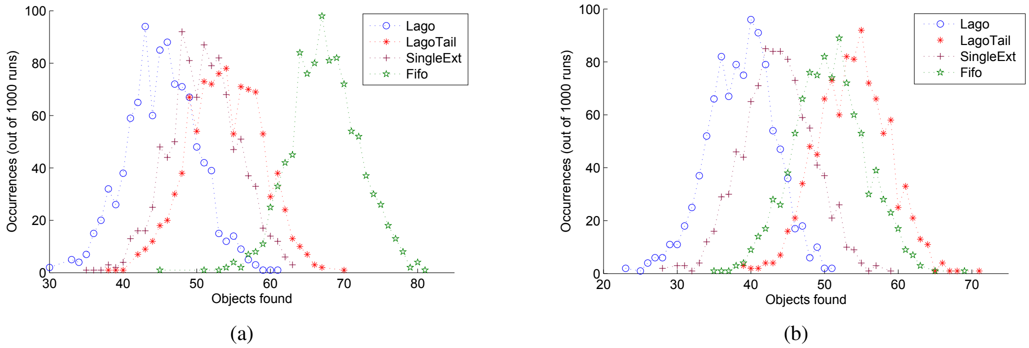

Fig. 8. The four algorithms with MINMIX descriptor in search of 100 objects limited to (a) half their C_{MM} time; (b) a same fixed mission time for all.

shorter plan of the best solution obtained by the LAGOTAIL algorithm. It would be interesting to pursue this point to determine at which problem sizes or mission durations FIFO is surpassed by the other algorithms.

As a final observation, the Gaussian curves found in figs. 7 and 8 arise because the distributions in figs. 5 and 6 satisfy the properties from which the central limit theorem follows. Particularly, for the two roughly uniform distributions, there are k objects to be uniformly found during mission total time. This follows a Poisson distribution with $\lambda = \frac{C_{\mathrm{MM}}}{k}$.

V. CONCLUSIONS

We have tested several auction techniques in a large simulated real scenario and in smaller random worlds. We have applied several criteria to optimize the MINMAX (mission time) and MINSUM (resource usage) costs. These criteria are linear combinations of both costs. Multiple runs have been performed to extract general trends, for several mission kinds like goal visiting or object finding.

The principal finding is that the MINMIX criterion, which combines equally time and resource costs, is very good in all cases. It is best for the time objective in a majority of occasions, better than any other time criterion for the resource objective, and very close to the MINSUM one for the resource objective. In the other hand, MINSUM is only good for its intended resource objective, being notably bad for the mission time. This finding strongly suggests that multi-robot teams should carefully justify the use of MINSUM auctions, due to its narrow applicability.

Of all tested algorithms, the only one with a patently different performance is the myopic FIFO one, which only considers one task ahead. Our experiments confirm early literature results where it is best suited for situations where tasks outnumber robots and task arrival is ongoing. For other kinds of problems, like finite maps and tasks, it is not a competitive alternative.

We have also observed that ongoing auctions after the first full allocation noticeably improve the mission plan. Thus, an initial bound-guaranteed auction (LAGO) followed by ongoing auctions of already assigned tasks (SINGLEEXT) would couple all the observed advantages.

Our experiments on object searching evidence that the MINSUM criterion is badly suited for this kind of mission in all cases. The greedy short-sightedness of FIFO makes it well-performing in the early mission stages, but as the time elapses the better plans of the other algorithms take over the lead.

Future work has the purpose of observe if these findings can be generalized to more situations. We intend to carry out further testing in more regular and human-made scenarios. Especially, we want to verify if the huge underperformance of the MINSUM descriptor for the mission time cost is also observed in these regular scenarios. Experiments with a wider range in the number of robots are also needed to learn more about the generality of the results found herein.

REFERENCES

[1] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Intl. J. of Robotics Research*, vol. 23, no. 9, pp. 939–954, September 2004.

[2] M. B. Dias, R. M. Zlot, N. Kalra, and A. T. Stentz, "Market-based multirobot coordination: a survey and analysis," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-05-13, April 2005.

[3] R. M. Zlot and A. T. Stentz, "Complex task allocation for multiple robots," in *ICRA'05*. IEEE, April 2005.

[4] N. Kalra, D. Ferguson, and A. T. Stentz, "Hoplites: a market-based framework for planned tight coordination in multirobot teams," in *ICRA'05*, April 2005, pp. 1182–1189.

[5] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-UAV context," in *ICRA'04*, May 2004.

[6] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[7] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[8] M. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multirobot task allocation," in *IROS'04*, 2004, pp. 698–705.

[9] A. R. Mosteo and L. Montano, "Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions," in *IROS'06 workshop on Network Robot Systems: Toward intelligent robotic systems integrated with environments*, 2006.

[10] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *ICRA'02*, 2002.

[11] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Concorde TSP solver." [Online]. Available: http://www.tsp.gatech.edu/concorde.html